

1. Environment

- A. OS : Windows 10
- B. compiler version : TDM-GCC 4.9.2 64-bit
- C. IDE : Dev-C++ 5.11

2. Methods or solutions

A. 從 txt 檔讀取陣列

- i. 用 `getline` 逐列讀取並存入一字串，並計算共有幾列，若字串為空，表示該陣列結束。
- ii. 每行讀取完皆將字串以空格為分割標的物，分割並存入 `vector`，最後將 `vector` 的 `size` 除以列數即可得行數。
- iii. 將 `vector` 內的元素逐一轉成 `int` 型別，存入另一動態記憶體配置的整數型別陣列 `arr`。

B. Young tableau insert

將要 insert 的元素 `ins` 取代 `arr` 中由上至下，由左至右第一個 ∞ ，放入後記錄放的位置 `(m, n)`，進入 `while` 迴圈，此時有四種情況，如下

- i. `m, n` 皆不是 0，則又有四種情況
 - 1. 若 `arr[m-1][n]` 及 `arr[m][n-1]` 皆大於 `ins`
 - A. 處理方法：選擇 `arr[m-1][n]` 與 `arr[m][n-1]` 兩者中較大的與 `ins` 互換，並須將 `ins` 換完後的位子記錄下來
 - B. 原因：若選擇較小的與 `ins` 互換，換完後該較小的元素會位於較大元素的下方或右方則需再做一次互換，故需選擇 `arr[m-1][n]` 與 `arr[m][n-1]` 兩者中較大的
 - 2. 若 `arr[m-1][n]` 大於 `ins`，`arr[m][n-1]` 小於 `ins`
 - A. 處理方法：將 `ins` 與 `arr[m-1][n]` 互換，並須將 `ins` 換完後的位子記錄下來
 - 3. 若 `arr[m-1][n]` 小於 `ins`，`arr[m][n-1]` 大於 `ins`
 - A. 處理方法：將 `ins` 與 `arr[m][n-1]` 互換，並須將 `ins` 換完後的位子記錄下來
 - 4. 若 `arr[m-1][n]` 及 `arr[m][n-1]` 皆小於 `ins`，因已符合 Young tableau 的規則，故 `while` 迴圈結束
- ii. `m` 等於 0，`n` 不等於 0
 - 1. 若 `arr[m][n-1]` 大於 `ins`
 - A. 處理方法：將兩者互換
 - B. 不須考慮換完後原 `arr[m][n-1]` 是否會大於 `ins` 原位子

的右方元素，因 ins 是逐步從右方及下方往左方及上方換，且每次皆選擇大於自己中的較大元素換，故不存在此種可能

2. 若 $\text{arr}[m][n-1]$ 小於 ins，因已符合 Young tableau 的規則，故 while 迴圈結束

iii. m 不等於 0， n 等於 0

1. 若 $\text{arr}[m-1][n]$ 大於 ins
 - A. 處理方法：將兩者互換
 - B. 不須考慮換完後原 $\text{arr}[m-1][n]$ 是否會大於 ins 原位於的右方元素，原因同 ii. 中的 1(B)
2. 若 $\text{arr}[m][n-1]$ 小於 ins，因已符合 Young tableau 的規則，故 while 迴圈結束

iv. m 等於 0， n 等於 0

1. 因 ins 為整陣列最小元素，且已移到 $(0, 0)$ ，符合 Young tableau 的規則，故 while 迴圈結束

C. Young tableau extract-min

將 $\text{arr}[0][0]$ 的值存入一整數型別變數，並將 $\text{arr}[0][0]$ 的值改為一極大值，並將 $(0, 0)$ 記錄在變數 $(\text{lastm}, \text{lastn})$ 中，則右方及下方元素必小於它，則進入一 while 迴圈，此時有兩種情況

i. $\text{arr}[\text{lastm}+1][0] \geq \text{arr}[0][\text{lastn}+1]$

1. 把 $\text{arr}[0][\text{lastn}+1]$ 與 $\text{arr}[\text{lastm}][\text{lastn}]$ 互換，並將 $\text{lastm}++$
2. 原因：選擇較小的原因同 Young tableau insert 時兩者皆大於 ins 時選擇大者互換

ii. $\text{arr}[\text{lastm}+1][0] < \text{arr}[0][\text{lastn}+1]$

1. 把 $\text{arr}[\text{lastm}+1][0]$ 與 $\text{arr}[\text{lastm}][\text{lastn}]$ 互換，並將 $\text{lastn}++$

iii. while 迴圈結束條件

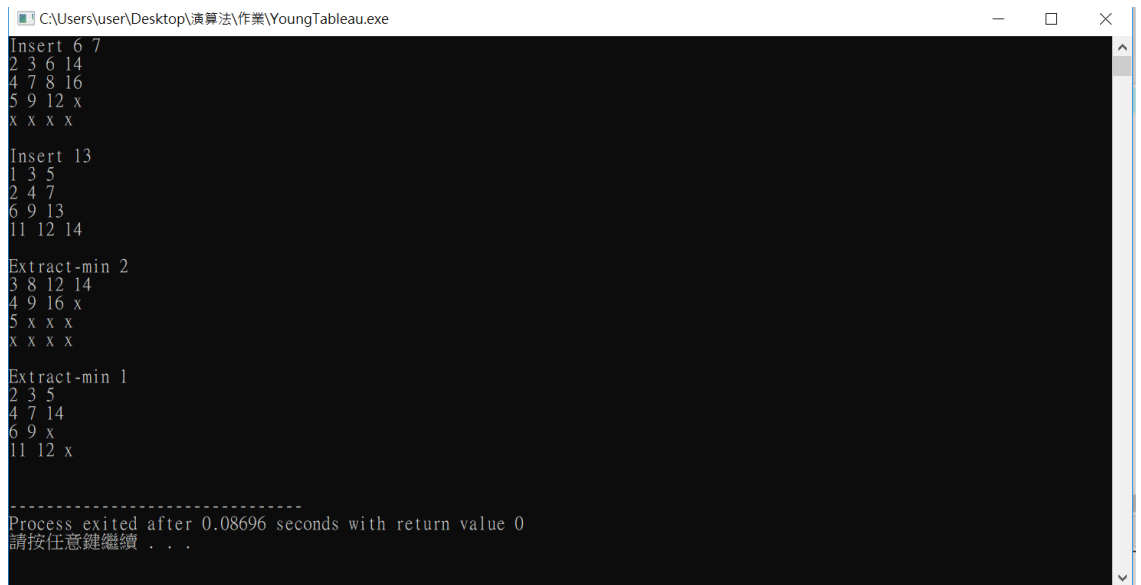
1. 若 lastm 不等於迴圈總列數-1，因陣列自 0 開始算，且 lastn 不等於迴圈總行數-1，則 $\text{arr}[\text{lastm}][\text{lastn}]$ 右方及下方元素皆等於它為該極大值
2. 若 lastm 等於迴圈總列數-1， lastn 不等於迴圈總行數-1，則 $\text{arr}[\text{lastm}][\text{lastn}]$ 右方元素等於它為該極大值
3. 若 lastm 不等於迴圈總列數-1， lastn 等於迴圈總行數-1，則 $\text{arr}[\text{lastm}][\text{lastn}]$ 下方元素等於它為該極大值
4. lastm 等於迴圈總列數-1，且 lastn 等於迴圈總行數-1

D. 寫入 txt 檔

i. 因需讀取 txt，故已引入標頭檔 fstream，則直接宣告 ofstream
即可輸出到 output.txt

3. Results

A. Dev C++ 編譯並執行後的結果



```
C:\Users\user\Desktop\演算法\作業\YoungTableau.exe
Insert 6 7
2 3 6 14
4 7 8 16
5 9 12 x
x x x x

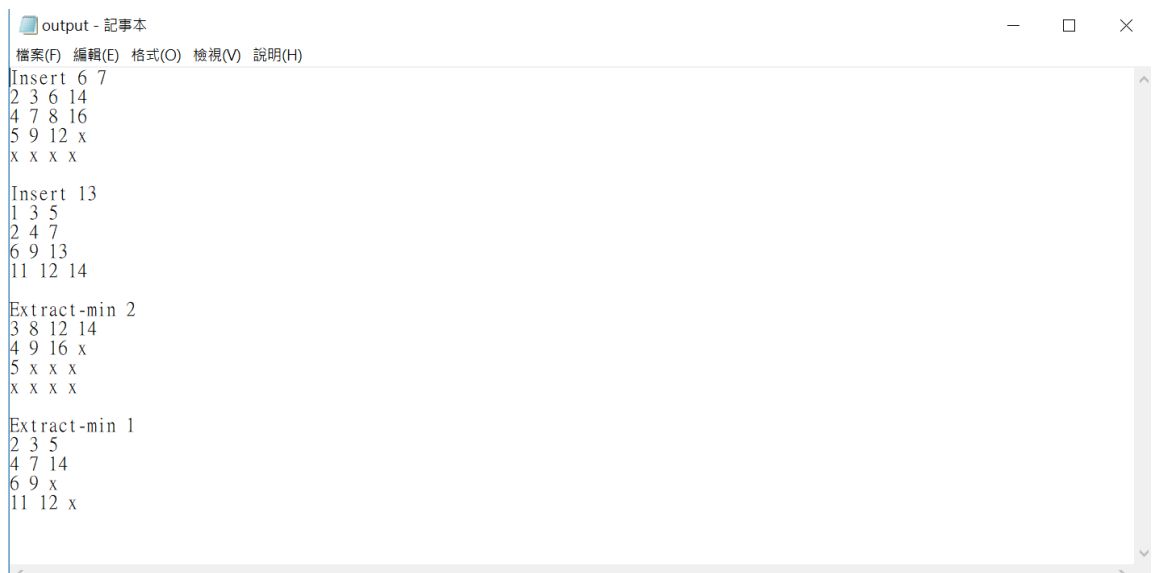
Insert 13
1 3 5
2 4 7
6 9 13
11 12 14

Extract-min 2
3 8 12 14
4 9 16 x
5 x x x
x x x x

Extract-min 1
2 3 5
4 7 14
6 9 x
11 12 x

-----
Process exited after 0.08696 seconds with return value 0
請按任意鍵繼續 . . .
```

B. Output.txt



```
output - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
Insert 6 7
2 3 6 14
4 7 8 16
5 9 12 x
x x x x

Insert 13
1 3 5
2 4 7
6 9 13
11 12 14

Extract-min 2
3 8 12 14
4 9 16 x
5 x x x
x x x x

Extract-min 1
2 3 5
4 7 14
6 9 x
11 12 x
```